

MCX

CMAA
Construction Management
Association of America

Member Communication Experience

Any views and opinions expressed in this article may or may not reflect the views and opinions of the Construction Management Association of America (CMAA). By publishing this piece, CMAA is not expressing endorsement of the individual, the article, or their association, organization, or company.

cmaanet.org

Software Issues Can Add to Project Delays and Challenges

Written by: Mariusz Wiechec, Director, HKA Global, LLC

An anticipated rise in infrastructure construction forecasted in early 2022 has not been as robust as the industry hoped, and construction officials are cautious about what 2023 will bring. Ongoing supply chain issues, spikes in the costs of construction materials, labor shortages across all trades, and the threat of inflation have dampened the optimism expressed last year.

While construction funding will continue to be rolled out in the United States throughout this year as part of the Infrastructure Investment and Jobs Act (IIJA), officials warn that rebuilding U.S. infrastructure still may face strong headwinds over the next several months.

Considering that a majority of today's infrastructure projects rely on new technology for their functionality — an industry-wide reality that did not exist a generation ago — the stakes for timely, on-budget completion can rise even higher. And those higher stakes can lead to concurrent increases in contentious delay-related disputes and litigation among stakeholders.

Technology's Rise

One of the components often overlooked in infrastructure projects is the development of software for safe and smooth operation. Software development is becoming an increasingly crucial component of construction projects and will only become more prevalent in the future.

Technology has become integral to the successful operation of nearly every type of construction. Indeed, it can be as important to a project's makeup as steel, bricks, and mortar. As a result, the construction industry has had to adjust and pivot from simply delivering infrastructure to delivering comprehensive functionalities. However, a firm grasp of technology — and its inherent promise and limitations — has lagged well behind its ubiquitous implementation.

This lag isn't a new concern. Nearly 60 years ago, in 1965, Intel Corporation co-founder Gordon Moore predicted that technology development would soon outpace the number of people embracing it, a prediction that has now been accepted as Moore's Law¹. As shown on the chart below, Moore's prediction was accurate², and various studies of technological growth show that this exponential pace of progress is also true for other features of computer engineering.³

¹ Gordon E. Moore - "Cramming more components onto integrated circuits", 1965

² Karl Rupp – "40 years of Microprocessor Trend Data"

³ <https://ourworldindata.org/technological-progress>

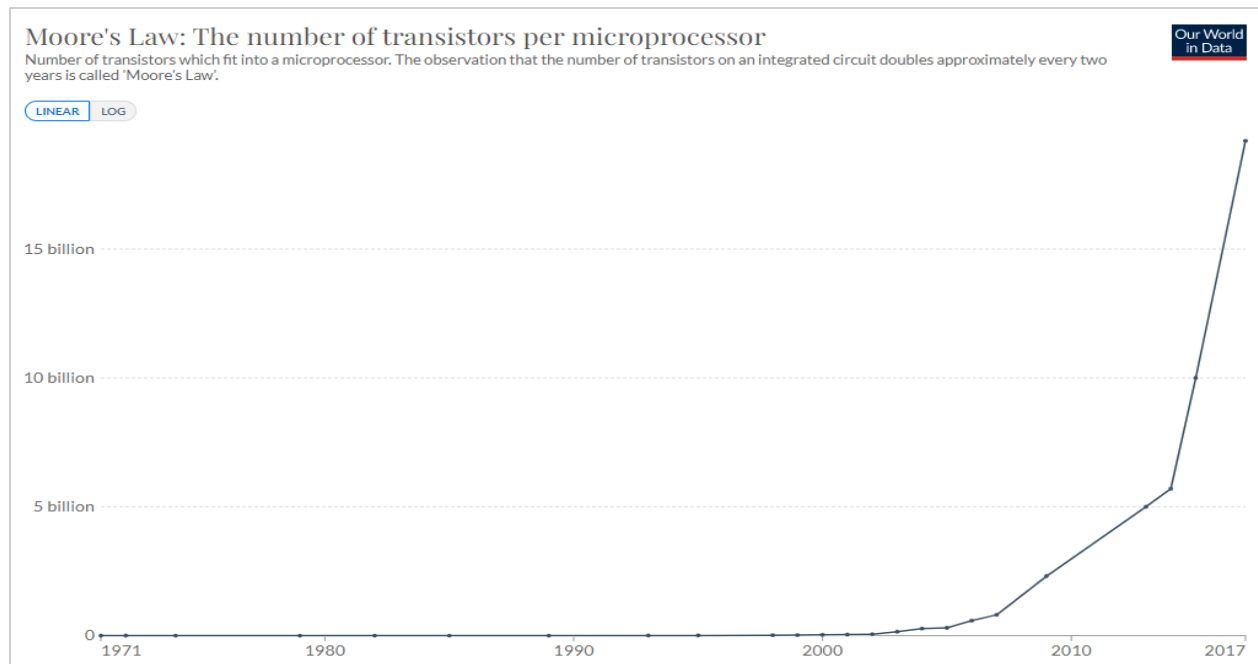


Figure 1 – Moore's Law – Exponential growth of transistors in microprocessors illustrates a broader truth about the technological progress of computer engineering.⁴

Innovation Versus Tradition

The concept of continuous technological innovation is somewhat of an outlier, however, in civil and structural engineering, where it can be safely assumed that the technology used today will likely be very similar and fit for purpose ten 10 years from now.

The exponential growth of computer technology⁵ necessitates a different approach to managing software development projects, employing more agile and less change-averse techniques. Unfortunately, these techniques are at odds with a traditional, rigid project management approach to construction projects, in which change is often difficult to manage.

This discrepancy between the rapidity of technological evolution and the traditional, rigid approach to project management makes it particularly challenging to manage projects that meld both civil/structural and software development aspects. And, because major infrastructure projects often are run by professionals with traditional construction management backgrounds, supporting systems such as software may not be foremost in their minds — or in their project documents and delivery plans.

Polls of industry professionals seem to bear out this hesitancy to fully embrace technology. According to a recent poll by GlobalData⁶, nearly half of all respondents said they feel that investing in technology promises little financial gain in the short term. According to GlobalData's "Trend Insight: Technology in Construction, 2021" report, construction companies also cited a lack of financial resources allotted for technological innovation (36%), a lack of sufficiently skilled labor (34%), and a lack of awareness of new technology (28%).

⁴ <https://ourworldindata.org/technological-progress>

⁵ <https://ourworldindata.org/technological-progress>

⁶ <https://www.designbuild-network.com/comment/tech-adoption-in-construction/>

“Familiarity Bias” and “Proximity Bias”

Innate biases, coupled with years of practice, may account for the construction industry’s collective reticence surrounding technology and tech-centered projects.

As construction professionals, we have an inherent bias when assessing the importance and risks of the tasks at hand. We tend to gravitate toward what we “know” and ignore what we do not, a trait that I am labeling as “familiarity bias.” Construction professionals know and understand construction. It’s a process with which they’re comfortable and “familiar.” As a result, they may be more actively engaged with the “bricks and mortar” aspects of the project than those aspects they know little about.

As humans, we also tend to focus on the things that are right in front of us, training all of our attention on the task at-hand, without consideration for what comes next — a phenomenon I’ll call “proximity bias.”

Typically, infrastructure projects are dominated at their beginning stages by heavy civil engineering design activities, followed by construction. Software needed for the project’s functionality is likely being developed concurrently, but that development often occurs in the background; not getting necessary “front-and-center” attention until it’s time for testing and implementation.

Both familiarity and proximity biases can cause a misalignment of priorities or a lack of attention to software development, which can result in delays and cost overruns as well as a scarcity of documentation that may prove crucial if disputes or claims arise.

Case Study - Crossrail

Crossrail was a railway construction project located in London and its outskirts. For more than a decade, it was a flagship infrastructure project in the U.K. When construction began in 2008, Crossrail was touted as the largest infrastructure project in Europe, with an initial budget of roughly GBP 17 billion (circa U.S. \$23 billion).

In its first few years, the Crossrail project was centered around the excavation of tunnels. Everyone understood that 26 miles of tunneling in central London would be an extremely challenging engineering feat, with BBC documentarians calling it “Urban Heart Surgery.”⁷

Tunneling works started in 2012, and while the 2014 National Audit Office (NAO) Report highlighted potential risks to project delivery, the overall tone of the messaging was positive, highlighting that: *“Crossrail Limited has achieved most or all milestones set for each six-month period on time.”*

The report further stated that there was an estimated 78-percent probability of Crossrail’s central section opening on time in December 2018.

Five years later, the tone of the NAO Report was less positive. Although impressive and by no means easy, tunneling did not prove to be the biggest challenge. Tellingly, the project started experiencing significant delays in 2015, when the tunneling scope was finished. Multiple problems plagued the delivery. However, the perceived critical delays were still related to infrastructure development (constructions of stations and the tunnel lining).

⁷ The Fifteen Billion Pound Railway - Urban Heart Surgery - https://www.youtube.com/watch?v=IrLNJi2Zffs&t=11s&ab_channel=EngineeringProjects

To mitigate these delays, the project team decided to overlap construction with dynamic testing. This put enormous pressure on the delivery team. The testing could not be conducted efficiently, however, as it was discovered that *“the train and signaling software had not been developed to the required level.”*⁸

Due to software development issues, *“few meaningful results could be acquired,”* and challenges with completing the testing effort *“took any spare time and space from construction workers on-site,”*⁹ causing further delays and cost increases.

It is important to note that, even if the project team had delivered the infrastructure scope on time, the underdeveloped and delayed software would have prevented successful testing of the entire system. While the software development was not the most significant issue causing delays to the delivery of Crossrail, it played a role — and an unexpectedly significant one.

Recently, Mark Wild, current CEO of Crossrail, stated in an interview, *“Crossrail is immensely complex, digital asset (...) more akin to a big IT program rather than infrastructure program.”*¹⁰ It was not always perceived as such.

Case Study - Electrical Substation

Projects do not need to cost billions of dollars to experience debilitating software issues. For example, a national electricity company hired an experienced general contractor to design, supply, deliver, install, and commission new equipment for a substation to support area power infrastructure improvements.

Apart from its civil, structural, and mechanical scopes, this \$20-million project also included the development of software to operate the newly installed substation equipment.

The project’s critical path delineated a fairly typical construction scope: Design → Civil and Structural Construction → Mechanical and Electrical Scope → Commissioning.

However, there was also another, separate path, which involved the development of operating software to be installed on control panels in the substation. The two paths converged at the commissioning stage.

⁸ NAO 2019 – Completing Crossrail – May 3, 2019

⁹ NAO 2019 – Completing Crossrail – May 3, 2019

¹⁰ The Smith School – “Are rail projects a nightmare to deliver? Lessons for successful delivery” – September 14, 2021

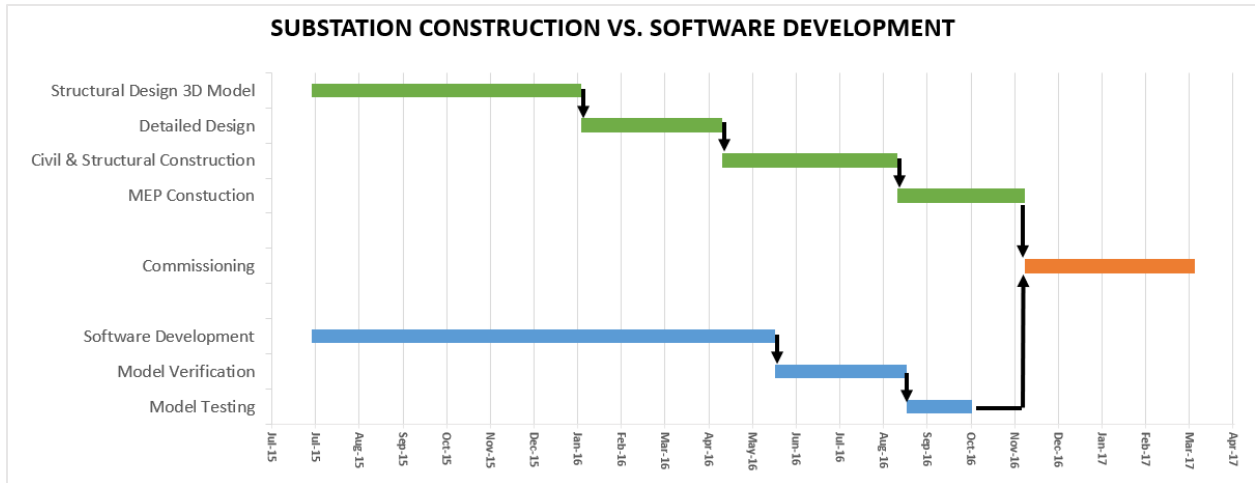


Figure 2 – Construction and Software Development paths (Illustration by Author)

As in Crossrail, for the first couple of years of the project, the project team focused on completing the civil scope while the software development was progressing in the background. For this project, the owner was responsible for providing a software architecture environment (or a network model) within which the newly developed software was to operate.

Ultimately, the lack of timely development of well-structured software architecture became a root cause of the delays on the project. Had the software development scope been better understood and received more attention from both the owner and the contractor earlier in the project, the completion of the works likely would not have been delayed. Instead, severe problems with software development did not become apparent to the broader project team until shortly before commissioning, when it was too late to create a viable mitigation plan.

Case Study - Positive Train Control Across the U.S.

In 2008, the U.S. Congress passed the Rail Safety Improvement Act of 2008,¹¹ requiring all Class I railroads' main lines that transport either hazardous materials or commuters to implement a Positive Train Control (PTC) system. Positive Train Control is a communication-based and processor-based technology that is designed to prevent train collisions, derailments, and other types of train accidents. Initially, the Act specified that PTC systems be fully implemented by the end of 2015. Congress extended its deadline to 2018, and then to 2020, when it became clear that most rail agencies needed more time.¹²

PTC implementations involve multiple stages: installation of wayside equipment/hardware along the tracks (including wayside interface units, communication base-station towers, signaling and highway crossing equipment, and fiberoptics), onboard train control equipment and computer systems, and back-office control center servers and equipment. PTC installation also requires a multi-stage testing and safety certification process. Notably, the installation necessitates the development of software

¹¹ The Rail Safety Improvement Act of 2008, Pub. L. No. 110-432, div. A, 122 Stat. 4848 (2008)

¹² GAO-19-135T – Testimony Before the Committee on Commerce, Science, and Transportation, U.S. Senate – Positive Train Control - October 3, 2018

allowing for automated train control operations and “communications” between the various components of the whole system.

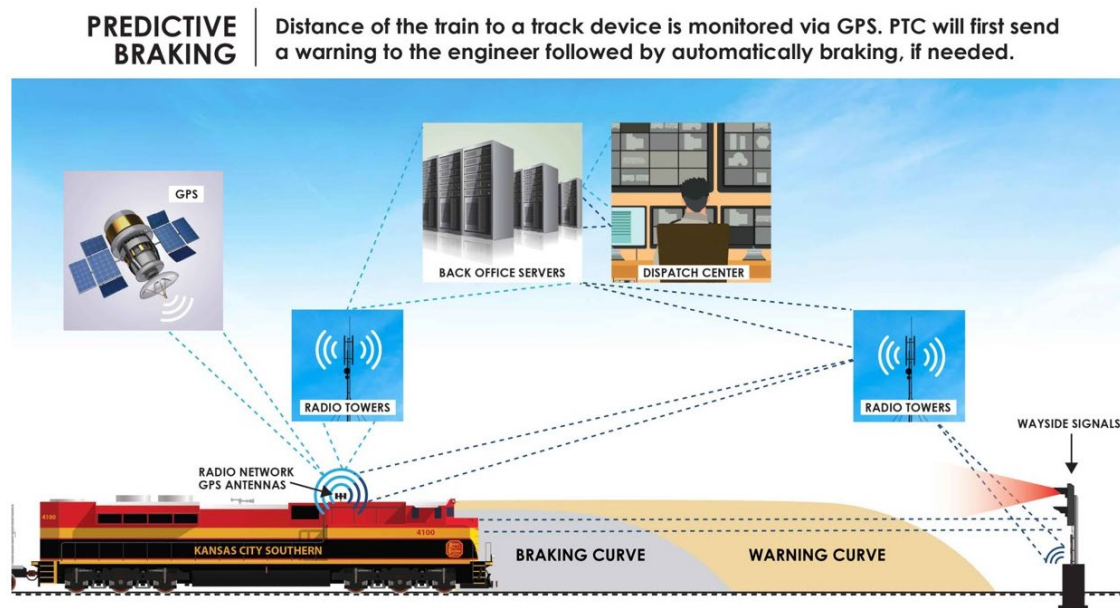


Figure 3 – Schematic representation of PTC system’s major components and interconnections.¹³

As with the Crossrail and Electrical Substation projects, PTC developments on railway systems in the U.S. have experienced multiple delays with similar root causes.

In its 2018 report,¹⁴ produced two years before the final PTC completion deadline, the U.S. Government Accountability Office (GAO) cited software issues as one of its concerns, but not the primary one. It is important to note that, in 2018, many railroads were at the early stages of their PTC testing program and had only begun to uncover the extent of software issues at this converging point. According to the report, “*nine passenger railroads reported encountering challenges related to maturity of the PTC software systems, such as working through software bugs or defects during testing.*”

A year later, the updated GAO report¹⁵ paints a different picture: “*31 of 37 railroads said software issues were a major or moderate challenge.*” As the testing progressed, many project owners discovered that the PTC software was not ready for the subsystem and system testing. Resolving the issues took longer than anticipated. “*Software issues are more acute now because as the 2020 deadline nears, less time remains to address these issues and associated delays,*” the report states.

As described earlier, the rate of technological progress for computer sciences vastly outpaces other engineering sectors. Of course, the rail agencies, the owners of the PTC upgrade projects,

¹⁴ GAO-19-135T – Testimony Before the Committee on Commerce, Science, and Transportation, U.S. Senate – Positive Train Control - October 3, 2018

¹⁵ GAO-19-693T - Testimony Before the Committee on Commerce, Science, and Transportation, U.S. Senate – Positive Train Control – July 31, 2019

understandably wanted to utilize the newest available technologies. However, as the software was constantly evolving, the potential for scope creep increased and the appropriate management of the software development became cumbersome.

Moreover, as opposed to traditional linear planning of construction scope, the software development process is, by its nature, cyclical and non-linear. Each functionality needs to be developed, tested, debugged, and retested. This process repeats at subsystem and system levels until the software is stable and functional.

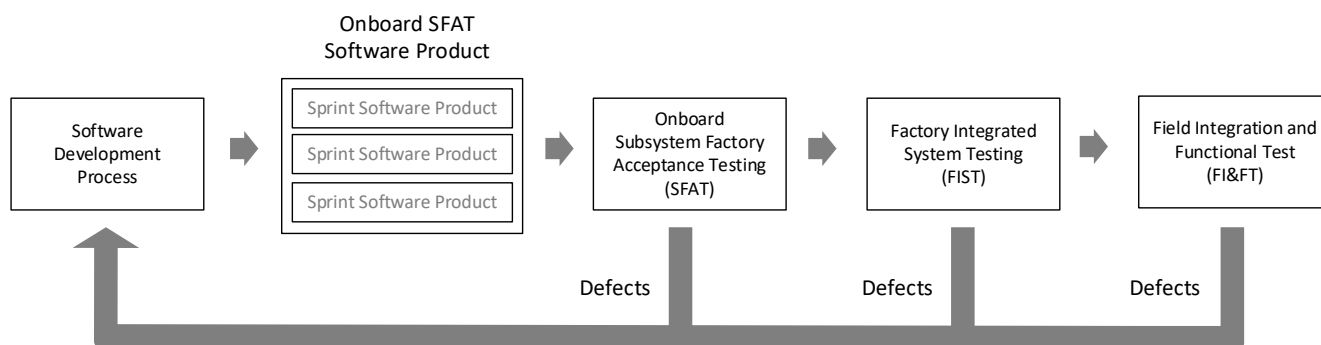


Figure 4 – Cyclical process of onboard software development and testing process on a PTC project.
(Illustration by Author)

For many PTC projects, in fact, the completion of software development has turned out to be a “moving target.” It has become apparent that without clearly defined functionalities for design freeze and a well-developed change procedure that takes into account differences between agile and traditional project management and planning methods, software development can quickly become a significant issue.

In the interviews conducted for the GAO 2019 report, some railroad representatives stated that “they had no control over this process [software development], as they must rely on the vendor to provide reliable software.”¹⁶ The report further suggests that “Representatives from this railroad also noted that resolving software issues is often not entirely within a railroad’s control due to the need for vendor support, in contrast to some earlier challenges leading up to the 2018 deadline, where, for example, the railroad itself had more control as it was installing equipment and could more clearly track progress.”¹⁷

Case Study – SEPTA’s On-Time PTC Technology Implementation

The Southeastern Pennsylvania Transportation Authority (SEPTA) was one of a handful of agencies throughout the U.S. to operate all of its trains with PTC ahead of the federal deadline. All SEPTA regional rail trains, operating in both SEPTA and AMTRAK territory, were PTC-compliant as of May 1, 2017.¹⁸ SEPTA, one of the nation’s largest transit agencies, serves Philadelphia and its five surrounding counties, with regional rail service extending as far north as Trenton, New Jersey, and as far south as Newark, Delaware. SEPTA also is one of only two U.S. transit agencies to operate all five types of on-land transit

¹⁶ GAO-19-693T - Testimony Before the Committee on Commerce, Science, and Transportation, U.S. Senate – Positive Train Control – July 31, 2019

¹⁷ GAO-19-693T - Testimony Before the Committee on Commerce, Science, and Transportation, U.S. Senate – Positive Train Control – July 31, 2019

¹⁸ <https://www.hka.com/positive-train-control/>

vehicles: regional commuter trains, heavy rapid transit (subway/elevated) trains, light rail vehicles (trolleys), trolleybuses, and motorbuses.¹⁹

Merging new PTC technology with legacy systems can be difficult, said Stephen J. Malaszecki, executive vice president at Envision Consultants, Ltd., a subcontractor on PTC projects in Philadelphia, New York and New Jersey. “Getting new technology to fit — and work well — with existing hardware isn’t easy, and it requires extensive testing to get it right and to pull everything together,” he said.

Retrofits become even more challenging if rail track is shared by more than one railroad. “When more than one agency is involved, there are always issues, especially when you’re dealing with track that’s used by both freight and passenger rail, for example,” said Malaszecki, who has nearly 40 years of experience in scheduling and project controls.

Ideally, systems testing should occur at intervals throughout a project — rather than at the end — and that testing should be allocated ample time in the schedule, Malaszecki said.

“When developing a schedule that includes equipment or systems testing, it is essential to review the detail of the project and determine the earliest time at which you can begin testing. Most projects place just one or two activities at the end of the schedule to reflect testing, but there are many instances where testing can begin much earlier,” he explained. “On a bus rapid transit project, for example, which usually involves miles of construction as stations are built consecutively, testing can begin on a station-to-station basis and proceed as construction moves forward. Then, final systems testing occurs when the construction is complete.

“When testing is scheduled and conducted incrementally, you can fix problems as they arise. If there’s an issue with the first test, for example, and there’s something in the system that is causing the discrepancy, you can fix it and then proceed, lessons learned, to the next station,” Malaszecki continued.

Ensuring that you’ve got the appropriate equipment for testing well before it begins also is important, Malaszecki added. “Testing may require its own unique equipment,” he said.

Paying attention to the role of testing in the project schedule and conducting it as early as possible can help save both time and costs later, Malaszecki emphasized. He and his team recently were able to reduce a project duration by six months by insisting on early testing.

“If you get started early, you can resolve problems early,” Malaszecki said.

In three of the four examples above, the lack of understanding or effective management of the software development process contributed to the project delays. With a heavy focus on the more tangible and typical construction tasks in the early stages of projects, the software development and testing scope did not appear to become a top priority until much later in the project. In each of the cases, software issues became a point of discussion only at the testing stage (or shortly before) when project teams had limited opportunities to mitigate impacts.

Malaszecki agreed. “You have to sit down and come up with a mutually derived plan that works for everyone, especially on projects that involve technology,” he said.

¹⁹ <https://www.hka.com/positive-train-control/>

Planning and communication are essential to that plan, Malaszecki added. “A highly detailed, consistently applied communication plan that is tied to the project schedule and includes each member of the project team — internal staff, outside consultants, contractors, subcontractors, and manufacturers — is extremely important,” he said.

Focus on Both Systems *and* Structures

In addition to implementing education and training to elevate understanding of technology and help change mindsets, there are leading practices that can help to prevent software development issues from derailing infrastructure projects:

- **Consistent, Interdisciplinary Planning and Scheduling.** It often seems as if software developers and construction professionals speak different languages. This becomes apparent in discussions about scheduling concepts. “Gantt Charts” and “finish to start logic ties” might not mean much to a software developer, just as “product roadmaps” or “blockers” may not sound familiar to a construction scheduler. However, the reality is that some of the concepts are very similar. In fact, the terms above mean *more or less* the same thing.

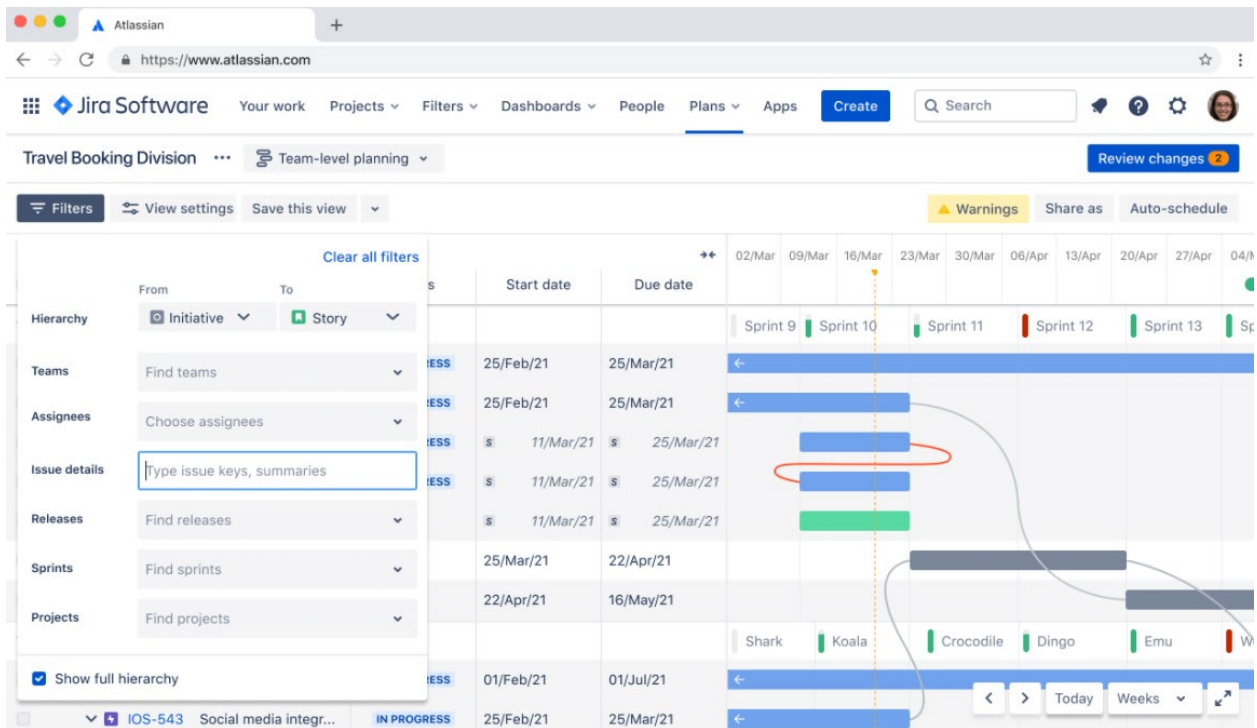


Figure 5 – ‘Roadmap’ developed in a popular software development tool – JIRA²⁰ heavily resembles construction projects “Gantt Charts.”

To bridge this gap, the software development plan must be well represented in the overall project infrastructure development schedule. The schedule also should account for a cyclical, non-linear approach to the software development process. The planner should carefully

²⁰ <https://www.atlassian.com/software/jira/features/roadmaps>

establish a planned duration of this process considering risk exposure levels. This can only be achieved if the planners and schedulers have sufficient understanding of the processes underlying the development and integration of software.

Also, the level of detail should allow for successful progress tracking of all tasks. Far too often, overall project schedules represent software tasks as single milestones (*“Software Delivered on date,”* for example, does not allow for good tracking of the progress). A good representation of software development progress in the project schedules also can be invaluable if delay analysis is needed. Most of the recognized delay analysis approaches rely heavily on the critical path method (CPM) and use of the official, approved schedule updates. If the software development is poorly represented in a project schedule, proving that it caused a critical delay to the completion may be very cumbersome.

- **Early Reporting and Data-Keeping.** Software development needs to be a meaningful part of project discussions early on. The functional software is typically only required at the latter, testing stages of an infrastructure project. With software being developed concurrently with more traditional construction tasks, and given the typically chaotic nature of construction projects, it is easy to ignore the software development scope. By the time the software is truly needed on a project, it is usually too late to mitigate delays. Therefore, much more attention and progress reporting are required early on. For example, software development can have a dedicated section in Periodic Progress Reports to highlight both progress and issues that need to be addressed. This reporting also can be crucial in a claim scenario when good project data is paramount.
- **Efficient Risk Management.** Risk Management efforts on construction projects often fall short due to “proximity bias.” We tend to focus on what’s right in front of us, rather than think beyond that and incorporate into our analysis things that aren’t currently on our “to-do lists.” As mentioned, functioning software is often needed toward the end of the project; therefore, it often takes time before it lands on the “to-do list” of decision-makers. By the time it does, risk mitigation options are limited. Therefore, project teams should consider software development risks early in the planning process.

It is important to note that all recommendations and suggested improvements will be fruitless without a major “re-think,” coupled with a change of approach in how infrastructure projects are managed. The U.K.’s Institution of Civil Engineers recognized the issue in its new recommendations:

“The dominant leadership and delivery model for infrastructure projects has not evolved to reflect these profound changes. Delivery remains in the hands of traditionally trained engineers working within organisations using long-established construction industry methods. The consequence of this conservatism is an increasing number of signature projects that are delivered behind schedule, beyond the cost estimate and that fail to meet the public’s expectations.”²¹

Malaszecki feels that the tide will change, especially as younger generations that have historically embraced new technologies advance in the industry. “It will happen, one project at a time,” he said.

²¹ Institution of Civil Engineers – “A systems approach to infrastructure delivery”

<https://www.ice.org.uk/engineering-resources/briefing-sheets/a-systems-approach-to-infrastructure-delivery-2>

Today, integrated, digital deliverables can be more important for end-users and owners than a “structure.” To address related risks, construction management’s focus must shift from traditional civil, structural, and mechanical scopes to a broader, all-encompassing delivery of functioning systems as part of smart infrastructure. “Sticks and bricks” should no longer be the sole focus of infrastructure upgrades. Rather, all system components should be treated as crucial to overall success, with software development high on the list.

###

This article presents the views, thoughts or opinions only of the author and not those of any HKA entity. While we take care at the time of publication to confirm the accuracy of the information presented, the content is not intended to deal with all aspects of the subject referred to, should not be relied upon as the basis for business decisions, and does not constitute professional advice of any kind. This article is protected by copyright © 2023 HKA Global, LLC.



Written by: Mariusz Wiechec, Director, HKA Global, LLC

HKA is a leading global consultancy in risk mitigation and dispute resolution. The firm’s multidisciplinary services include expert, claims and advisory services for the capital projects and infrastructure sector, forensic accounting and commercial damages services for all types of contracts, and consulting services to support companies working on U.S. Federal Government contracts. For more information about HKA, visit [hka.com](https://www.hka.com) and connect with us on LinkedIn (@HKAGlobal) and Facebook.